**INDEX**

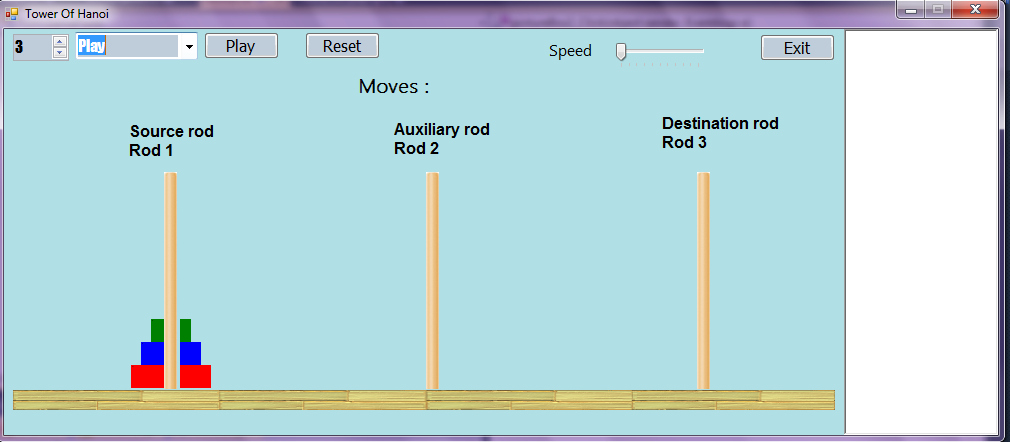
|  |  |
| --- | --- |
| **SR.No** | **Contents** |
| 1 | Introduction to project |
| 2 | System requirement |
| 3 | Entity Relationship Diagram |
| 4 | Rules of game |
| 5 | The requirements |
| 6 | The design |
| 7 | UI elements |
| 8 | Solve algorithm |
| 9 | Screenshots of game |
| 10 | Advantages of game |
| 11 | Conclusion |
| 12 | Biblography |

**INTRODUCTION**

* **Project Overview**:

This project contains a recursive solution for the Tower of Hanoi problem. This application is written in C# and UI is done using Windows Forms.

The Tower of Hanoi is a mathematical puzzle. It consists of three rows, and a number of disks of different size which can slide onto ant rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top. We have to obtain the same stack on the third rod.



**Hardware and Software Requirements**

* **Hardware**

1)Processor Minimum Pentium 4

2)RAM Minimum 1 GB

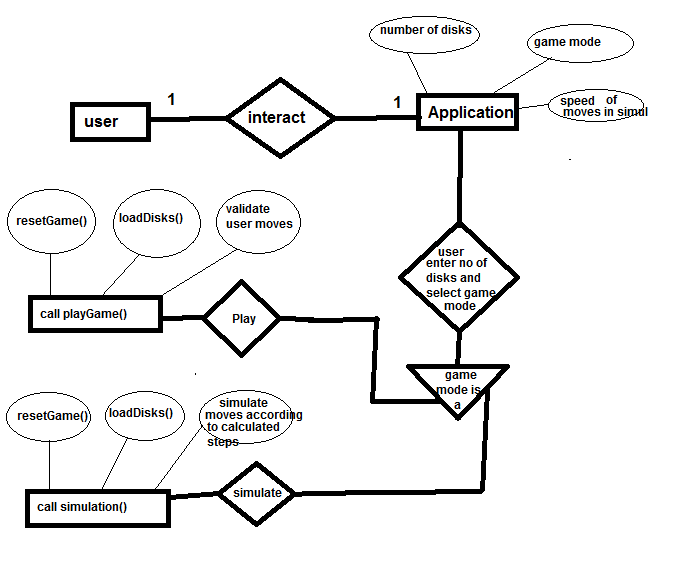
3)Hard Disk 80 GB

* **Software**

Following is the software package used to develop this application.

1. Software Package Microsoft Visual Studio 2015
2. Front end Windows Forms.
3. Back end C#

**Entity Relationship Diagram (ERD)**

****

* **Rules of game:**

Following are the rules to play the game:-

1. We can only move one disk at a time.
2. We cannot place a bigger disk on a smaller disk.
3. We have to move all disk from source rod to destination rod with the help of auxiliary rod (as shown in figure).

* **The requirements :**

1. A graphical representation, using Windows Forms, of the puzzle.
2. The user should be able to choose if they would like to use 1,2,3,4,5,6,7,8,9 disks\* in the puzzle. There will always be three poles\*\* present.
3. The application should allow only valid moves as defined by these rules: 1. You may only move one disk at a time. 2. You cannot place a bigger disk on a smaller disk.
4. The application must have a ‘simulate ‘ feature where the application will show the user the solution, step by step, for the selected number of disks.

* **The design :**

I wanted a clear separation between the UI and the backend. I created the SolveHanoi class with the sole purpose of working on the solution.

This SolveHanoi class has Tower() method which takes the number of disks and it recursively solve the solution and store the solution in List of steps.

Following is code for Tower() method of SolveHanoi class

public void Tower(int numberOfDisks, int src, int dest, int aux, List<Step> steps)

{

if (numberOfDisks == 1)

{

step.rod\_num = numberOfDisks;

step.dest = dest;

step.src = src;

steps.Add(step);

}

else

{

Tower(numberOfDisks - 1, src, aux, dest, steps);

step.rod\_num = numberOfDisks;

step.dest = dest;

step.src = src;

steps.Add(step);

Tower(numberOfDisks - 1, aux, dest, src, steps);

}

}

* **UI elements :**

The Game Form contains all the graphical components and severs as the driver for the application. I had used Label control as the base object for the disks and PictureBox control as the base object for Pole.

The Disk Label got the responsibility to move itself around. I had added click events to move disks from one pole to another

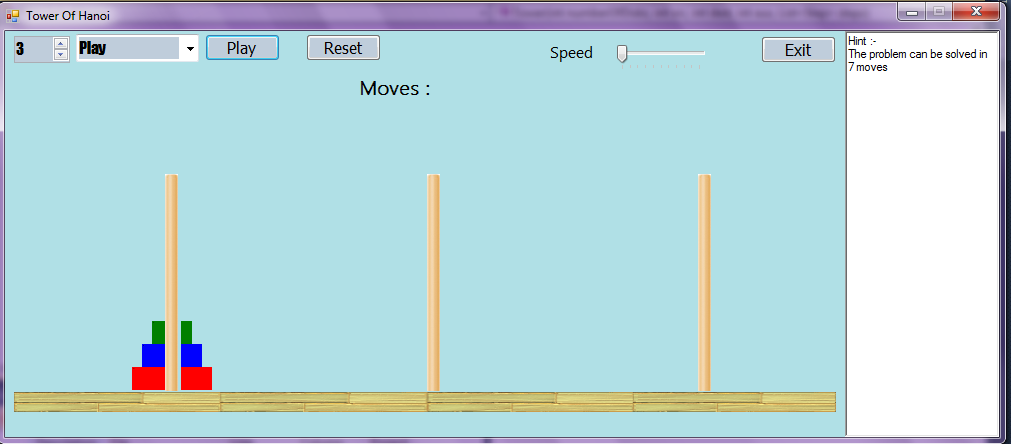
* **Solve Algorithm:**

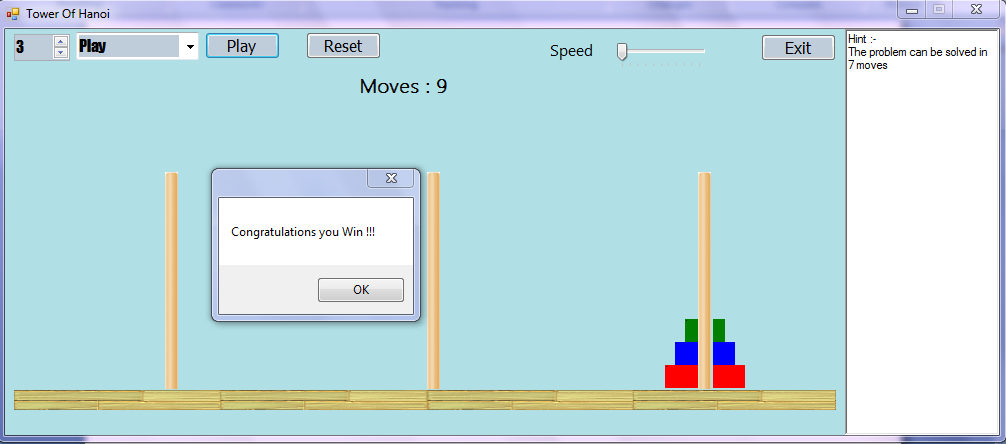
The algorithm used to solve the puzzle is a very simple recursive function. In the function each moves gets add to a list. This list gets used to solve the puzzle.

In the start state of game all the disks will be on the ‘start pole’. After executing all the moves in the list, all the disks should be on the ‘end pole’.

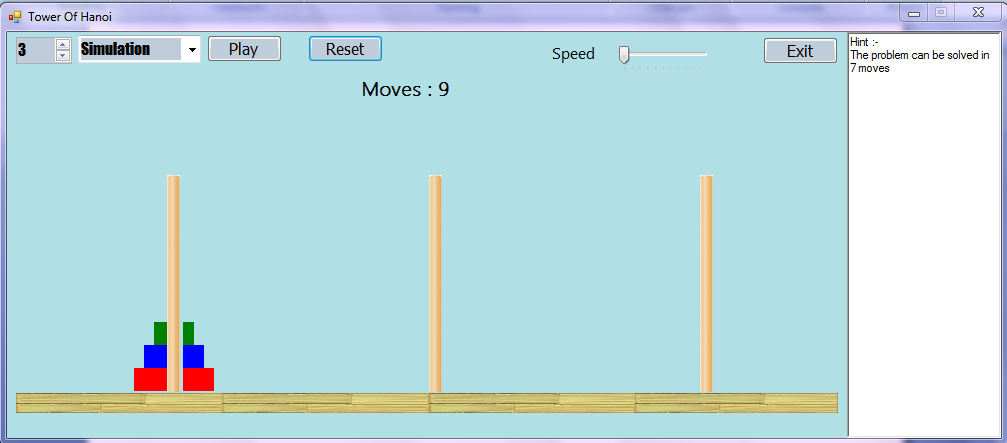
* **Screenshots of game :**

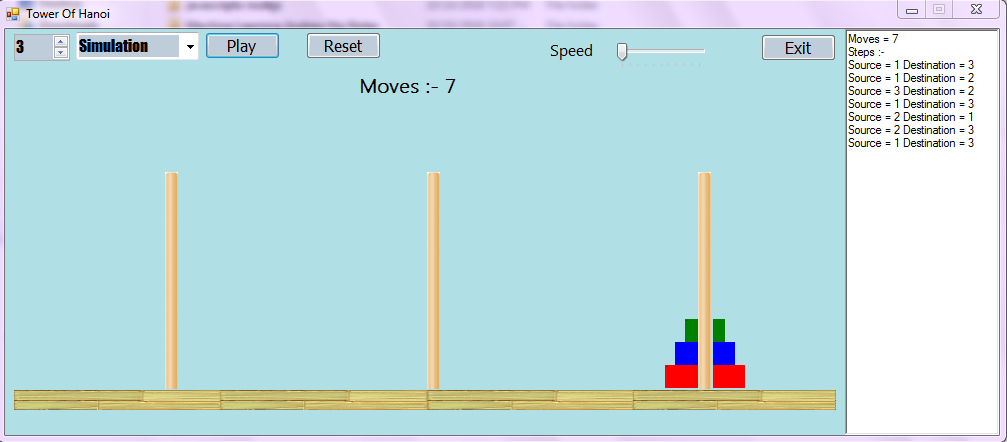
1. **When game in play mode.**

****

****

1. **When game in simulation mode.**

****

****

**ADVANTAGES OF PROJECT**

* **Simple and easy to play game.**
* **The game can be played on any low specification system.**
* **Easy user interaction.**

**DISADVANTAGES OF PROJECT**

* **Can only run on windows machine.**

**Conclusion:**

Working on project was good experience, I had understood the importance of planning and designing.

Developing the project has given me some experience on real time development procedure. By making this project I had learn more things about the project management. I had learned how to plan, how to design, how to create efficient algorithm, how to write efficient code, how to implement it, and how to test it.

**BIBLIOGRAPHY**

The following books and websites where referred during the analysis and execution phase of the project.

WEBSITE REFERRED:

<https://www.stackoverflow.com>

<https://www.github.com>

<https://docs.microsoft.com>

<https://www.geeksforgeeks.org>